

[illegible]

• • •  
• • •  
• • •  
• • •

```

RRRRRRRR      EEEEEEEEEEE      QQQQQQ
RRRRRRRR      EEEEEEEEEEE      QQQQQQ
RR              RR      EE      QQ              QQ
RR              RR      EE      QQ              QQ
RR              RR      EE      QQ              QQ
RR              RR      EE      QQ              QQ
RRRRRRRR      EEEEEEEEEEE      QQ              QQ
RRRRRRRR      EEEEEEEEEEE      QQ              QQ
RR      RR      EE      QQ              QQ
RR      RR      EE      QQ              QQ
RR              RR      EE      QQ              QQ
RR              RR      EE      QQ              QQ
RR              RR      EE      QQ              QQ
RR              RR      EEEEEEEEEEE      QQQQ      QQ
RR              RR      EEEEEEEEEEE      QQQQ      QQ

```



## DEFINITION FILE FOR ACL EDITOR COMPILATION

Version 'V04-000'

\*\*\*\*\*  
\* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY \*  
\* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. \*  
\* ALL RIGHTS RESERVED. \*  
\*\*\*\*\*

\* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED \*  
\* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE \*  
\* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER \*  
\* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY \*  
\* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY \*  
\* TRANSFERRED. \*  
\*\*\*\*\*

\* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE \*  
\* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT \*  
\* CORPORATION. \*  
\*\*\*\*\*

\* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS \*  
\* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. \*  
\*\*\*\*\*

++  
FACILITY: Miscellaneous utilities

## ABSTRACT:

These are the data structure definitions and random macros  
used to compile the ACL editor.

## ENVIRONMENT:

VAX/VMS operating system, user mode utilities.

--  
AUTHOR: L. Mark Pilant CREATION DATE: 7-Jul-1982 10:10

## MODIFIED BY:

V03-009 LMP0270 L. Mark Pilant, 29-Jun-1984 8:51  
Add definition for the control-C abort message.

V03-008 LMP0213 L. Mark Pilant, 24-Mar-1984 12:23  
Add support for locking and unlocking the object's ACL.

V03-007 LMP0193 L. Mark Pilant, 14-Feb-1984 12:44  
Move the journal and recovery file RMS structures to the  
common storage as they are referenced from multiple modules.

V03-006 LMP0181 L. Mark Pilant, 17-Jan-1984 10:02  
Add support for the SCHANGE\_ACL system service instead of  
direct ACP calls.

V03-005 LMP0172 L. Mark Pilant, 28-Nov-1983 12:11  
Numerous bug fixes, support for VT2xx terminals, and a  
session keystroke logger.

V03-004 LMP0138 L. Mark Pilant, 17-Aug-1983 8:59  
Misc fixes to prompt mode to cure ACCVIOs.

V03-003 LMP0124 L. Mark Pilant, 22-Jun-1983 11:37  
Change OWNER access definition to CONTROL.

V03-002 LMP0103 L. Mark Pilant, 20-Apr-1983 10:23  
Add support for the HIDDEN and PROTECTED ACE flag bits.

V03-001 LMP0074 L. Mark Pilant, 20-Jan-1983 12:02  
Add support for modifying the RMS journal bits if the  
corresponding ACE is added or deleted.

\*\*



! Local definition files.

REQUIRE 'LIBS:AEDLCLDEF';

! Declare VAX built in functions.

BUILTIN  
INSQUE,  
REMQUE;

! Macros to allocate and deallocate dynamic memory.

```
MACRO
  ALLOCATE (SIZE, ADDRESS) =
    BEGIN
      EXTERNAL ROUTINE      LIB$GET_VM : ADDRESSING_MODE (GENERAL);
      LOCAL VM_STATUS;
      VM_STATUS = LIB$GET_VM (%REF (SIZE), ADDRESS);
      IF VM_STATUS THEN C$FILL (0, SIZE, .ADDRESS);
      .VM_STATUS
    END
    %;

  DEALLOCATE (SIZE, ADDRESS) =
    BEGIN
      EXTERNAL ROUTINE      LIB$GET_VM : ADDRESSING_MODE (GENERAL);
      LIB$FREE_VM (%REF (SIZE), ADDRESS);
    END
    %;
```

! Macro to signal an error and save the worst case error condition.

```
MACRO
  SIGNAL (ARG) =
    BEGIN
      EXTERNAL ROUTINE      LIB$SIGNAL : ADDRESSING_MODE (GENERAL);
      IF .AED_L_FLAGS[AED_V_SCOPE]
      THEN
        BEGIN
          SCR$ERASE_PAGE (21, 1);
          SCR$SET_CURSOR (21, 1);
        END;
      LIB$SIGNAL (ARG %IF %LENGTH-1 GTR 0 %THEN, %REMAINING %FI);
      IF .AED_L_FLAGS[AED_V_SCOPE]
      THEN SCR$SET_CURSOR (.AED_B_LINE, .AED_B_COLUMN);
      IF (ARG AND ST$M_SEVERITY) NEQ ST$M_WARNING
      AND (.AED_L_WORSTERR AND ST$M_SEVERITY) LSS
        (ARG AND ST$M_SEVERITY) THEN AED_L_WORSTERR = ARG;
    END
    %;
```

! Shared message definitions.

```
$SHR_MSGDEF (AED, 277, LOCAL,
              (OPENIN, ERROR),
              (CLOSEIN, ERROR),
```

(READERR, ERROR)  
(WRITEERR, ERROR)  
);



! Define common storage structure.

PSECT OWN=AED\_COMMON(OVERLAY,ADDRESSING\_MODE(WORD\_RELATIVE),ALIGN(0));

OWN

AED_L_FLAGS	: SBBLOCK [4],	Useful flags
AED_B_OPTIONS	: SBBLOCK [1],	Qualifier option flags
AED_L_OBJTYP	:	Target object type code
AED_Q_OBJNAM	: SBBLOCK [DSC\$C_S_BLN],	! Object name descr
AED_L_WORSTERR	:	Worst error encountered
AED_L_PAGEWIDTH	:	Device line width
AED_L_PAGESIZE	:	Number of lines on the page
AED_B_COLUMN	: VECTOR [1,BYTE],	Current column position
AED_B_LINE	: VECTOR [1,BYTE],	Current line position
AED_B_SAVE_COL	: VECTOR [1,BYTE],	Last set column position
AED_B_SAVE_LIN	: VECTOR [1,BYTE],	Last set line position
AED_Q_LINETABLE	: SBBLOCK [12],	Input line queue head
		Note: Extra longword is necessary because of the way the string search loop is designed.
AED_L_CURACE	: REF SBBLOCK,	Address of current ACE
AED_L_FIRSTLINE	: REF SBBLOCK,	First line segment of ACE
AED_L_LASTLINE	: REF SBBLOCK,	Last line segment of ACE
AED_L_BEGINLINE	: REF SBBLOCK,	Address of first line in display
AED_W_INPUTLEN	: VECTOR [1,WORD],	Size of current ACE text
AED_Q_DEL_ACE	: SBBLOCK [8],	Deleted ACE line queue head
AED_Q_DEL_LINE	: SBBLOCK [DSC\$C_S_BLN],	Deleted line descr
AED_Q_DEL_WORD	: SBBLOCK [DSC\$C_S_BLN],	Deleted word descr
AED_B_DEL_CHAR	: VECTOR [1,BYTE],	Deleted character
AED_A_ACLBUFFER	: REF SBBLOCK,	Address of binary ACE
AED_Q_OUTLINE	: SBBLOCK [DSC\$C_S_BLN],	Output line descr
AED_W_OBJCHAN	: VECTOR [1,WORD],	Channel for ACL I/O
AED_W_TERMIN	: VECTOR [1,WORD],	Terminal input channel
AED_W_TERMOUT	: VECTOR [1,WORD],	Terminal output channel
AED_W_IOSB	: VECTOR [4,WORD],	I/O status block
AED_L_STATUS	:	Routine exit status
AED_B_FIELD	: VECTOR [1,BYTE],	Current field number
AED_W_FIELDBEG	: VECTOR [1,WORD],	Beginning position of field
AED_W_FIELDEND	: VECTOR [1,WORD],	End position of field
AED_B_ITEM	: VECTOR [1,BYTE],	Current item number
AED_W_ITEMBEG	: VECTOR [1,WORD],	Beginning position of item
AED_W_ITEMEND	: VECTOR [1,WORD],	End position of item
AED_B_ACETYPE	: VECTOR [1,BYTE],	ACE type (for prompting)
AED_W_JOURNAL	: VECTOR [1,WORD],	RMS journaling flags from header
AED_T_CURLINE	: SBBLOCK [512 + \$BYTEOFFSET (LINE_T_TEXT)],	Input line segment storage
AED_W_TOTALSIZE	: VECTOR [1,WORD],	Total size of all ACE segments

! Journal and recovery file RMS data structures.

JOURNAL_FAB	: \$FAB_DECL,	Journal file FAB
JOURNAL_NAM	: \$NAM_DECL,	Journal file NAM block
JOURNAL_RAB	: \$RAB_DECL,	Journal file RAB
JOURNAL_XABPRO	: \$XABPRO_DECL,	Journal file PROtection XAB
JOURNAL_BUFFER	: VECTOR [10, BYTE],	Storage for journaled keys

JOURNAL_INDEX,		!	Index into journal buffer
RECOVER_FAB	: SFAB_DECL,	!	Recovery file FAB
RECOVER_NAM	: SNAM_DECL,	!	Recovery file NAM block
RECOVER_RAB	: SRAB_DECL,	!	Recovery file RAB
RECOVER_BUFFER	: VECTOR [10, BYTE],	!	Storage for recovery record
RECOVER_INDEX;		!	Index into recovery buffer

PSECT OWN=\$OWNS(CONCATENATE,ADDRESSING\_MODE(WORD\_RELATIVE),ALIGN(2));



## ! External routines

## EXTERNAL ROUTINE

CLISGET VALUE	: ADDRESSING_MODE (GENERAL),	! Get qualifier value
CLISPRESNT	: ADDRESSING_MODE (GENERAL),	! See if qualifier present
LIB\$FREE VM	: ADDRESSING_MODE (GENERAL),	! Release dynamic memory
LIB\$GET VM	: ADDRESSING_MODE (GENERAL),	! Allocate dynamic memory
LIB\$PARSE	: ADDRESSING_MODE (GENERAL),	! General purpose parser
SCR\$DOWN SCROLL	: ADDRESSING_MODE (GENERAL),	! Scroll display down 1 line
SCR\$ERASE_LINE	: ADDRESSING_MODE (GENERAL),	! Erase a line of the display
SCR\$ERASE_PAGE	: ADDRESSING_MODE (GENERAL),	! Erase a portion of the display
SCR\$SET_CURSOR	: ADDRESSING_MODE (GENERAL),	! Set display cursor position
SCR\$SET_SCROLL	: ADDRESSING_MODE (GENERAL),	! Set display scrolling region
SCR\$UP_SCROLL	: ADDRESSING_MODE (GENERAL),	! Scroll display up 1 line

## ! External error message definitions

## EXTERNAL LITERAL

## ! Fatal error.

AED\$\_OBJLOCKED,

## ! Recoverable errors.

AED\$\_BADKEEP,  
AED\$\_LOCATERR,  
AED\$\_INIREADERR,

## ! Warning messages.

AED\$\_JOUWRITERR,  
AED\$\_JOUOPENOUT,  
AED\$\_JOUCLOSEOUT,  
AED\$\_RECREADERR,  
AED\$\_RECOPENIN,  
AED\$\_RECLOSEIN,  
AED\$\_BADUIC,  
AED\$\_BADGRPHEM,  
AED\$\_SYNTAX,  
AED\$\_BADTYPE,  
AED\$\_NOITEMSEL,  
AED\$\_MUSTENTER,  
AED\$\_INIOPENIN,  
AED\$\_INICLOSIN,  
AED\$\_DEFSYNTAX,  
AED\$\_NODELETE,  
AED\$\_NOMODIFY,  
AED\$\_NOHIDDEN,  
AED\$\_DUPLICATE,  
AED\$\_NOCOMBINE,  
AED\$\_NODEFAULT,

## ! Informational messages.

AED\$\_NOCTRLCHAR,

ACLEDTDEF.REQ;1

16-SEP-1984 16:47:25.72 <sup>N 14</sup> Page 8

AED\$\_NOTFOUND,  
AED\$\_CONTROL\_C.

! Success messages.

AED\$\_ACLUPDATED,  
AED\$\_NOCHANGE;



0002

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY